

Integrating Path Planning and Pivoting

Silvia Cruciani and Christian Smith

Abstract—In this work we propose a method for integrating motion planning and in-hand manipulation. Commonly addressed as a separate step from the final execution, in-hand manipulation allows the robot to reorient an object within the end-effector for the successful outcome of the goal task. A joint achievement of repositioning the object and moving the manipulator towards its desired final pose saves time in the execution and introduces more flexibility in the system. We address this problem using a pivoting strategy (i.e. in-hand rotation) for repositioning the object and we integrate this strategy with a path planner for the execution of a complex task. This method is applied on a Baxter robot and its efficacy is shown by experimental results.

I. INTRODUCTION

Several tasks in robotics include picking up objects or tools, followed by placing or using the grasped object. The problem of grasping an object according to its use is still an open challenge. Moreover, many times the desired grasp is impossible to achieve due to environmental constraints or robot's limits. When the desired configuration for the grasp is not achievable at first, the robot has to manipulate the object or interact with the environment to achieve its final goal. One solution for obtaining the desired grasp on an object is in-hand manipulation.

Differently from *regrasping* approaches that consist of picking up the object and placing it back multiple times until the desired grasp can be achieved [1], [2], in-hand manipulation keeps the object within the robot's end-effector while moving it towards the desired configuration. Accurately moving an object within the robotic hand or gripper falls in the category of *dexterous manipulation* problems [3].

Mimicking the human ability of precisely moving the fingers to manipulate an object requires an end-effector rich in *intrinsic dexterity*, i.e. a multi-fingered hand [4], [5] or a custom-made gripper tailored for the problem [6]–[9].

However, many commonly available robots (e.g. PR2, Baxter, Yumi) are only equipped with parallel grippers, which are simple to control and robust in the execution, but lack intrinsic dexterity. To enhance the poor intrinsic dexterity of these grippers the robots take advantage of the *extrinsic dexterity* [10]. The extrinsic dexterity exploits external forces, inertial forces, and contacts. The addition of these external supports allows the robot to perform dexterous manipulation tasks without the need for a dexterous hand.

In this work, we focus on an in-hand repositioning task that has to be executed together with the robot's motion.

Silvia Cruciani and Christian Smith are with the Robotics, Perception and Learning Lab, EECS at KTH Royal Institute of Technology, Stockholm, Sweden. {cruciani, ccs}@kth.se

This work was supported by the European Union framework program H2020-645403 RobDREAM.

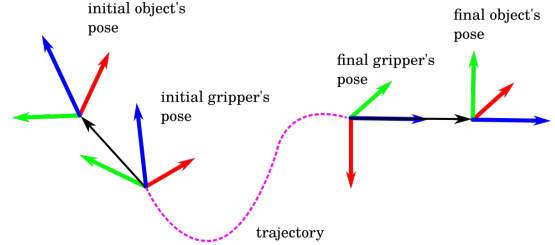


Fig. 1: A generic representation of task execution with in-hand repositioning. The gripper moves from the initial pose to the desired final pose, and at the same time the object's pose with respect to the gripper changes from the initial one to the desired one.

Most of the previous work on in-hand repositioning assumes a separate step for this dexterous task, before the robot starts using the object that it is grasping. We assume instead that after picking up an object, the robot starts moving towards the next pose, at which it needs to have the object in a different configuration. Fig. 1 shows a representation of this problem.

An example of this problem consists of a robot that has to pick up a screwdriver and use it to turn a screw. For turning the screw, the robot must have the screwdriver at a particular configuration inside the gripper, but this configuration is different from the one of the initial grasp. Instead of first performing a repositioning and then approaching the position of the screw, the motion of the robot towards the final configuration can be chosen so that it performs in-hand manipulation at the same time.

This methodology saves time with respect to one that splits the problem into two subtasks, such as first repositioning the object inside the gripper and then moving the robot to the desired configuration, or vice-versa. It also enhances the efficiency of the system as a whole; in fact, instead of imposing to the robot additional motions completely disconnected from the task, with the purpose of generating the dynamics necessary to the extrinsic dexterity solution, this method exploits the dynamics that would be generated anyway during the execution.

Among the different kinds of possible in-hand repositioning, we specifically address the problem of pivoting an object between the two fingers of a parallel gripper so that, once the robot has approached the final pose of the overall task, the grasped object rotates to the desired orientation within the gripper itself.

For the pivoting action, we generalize the method that we presented in [11] to be used on a more generic set of problems. We show how to integrate the pivoting action into a path-planner to obtain a trajectory that solves both the problem of moving the robot manipulator to the desired

configuration and of pivoting the object to the desired angle.

II. RELATED WORK

To the best of our knowledge, the problem of in-hand repositioning via pivoting has always been addressed as a separate step in the overall execution. Therefore, the execution of a complex task requires multiple independent steps, one of which is dedicated to in-hand repositioning. Conversely, we propose a method to achieve this repositioning together with the robot's motion towards the final goal pose. This joint execution allows the robot to save time when performing complex tasks.

To address this problem as a whole, a first possibility is on-line trajectory optimization, or model predictive control. However, this method has known issues with discontinuous dynamics and contact phenomena, in addition to the requirement for fast feedback and high-frequency controllability of the system.

It is also possible to formulate our overall problem as an instance of a kinodynamic motion planner [12]. More specifically, the pivoting point can be considered as a passive joint, making the overall system an underactuated system, and it is possible to adapt kinodynamic planners designed for underactuated robots to this problem [13]. However, these planners are slower than purely geometric planners because they must handle additional constraints on kinematic and dynamic level. Moreover, accurate models of both the robot's dynamics and the contact dynamics are required.

Therefore, we prefer to integrate a pivoting method with a simple geometric planner and generate the timing law of the trajectory according to the actions required for the successful outcome of the pivoting task.

Since pivoting is considered an extrinsic dexterity problem, previous solutions exploit external and inertial forces, contact surfaces and friction control.

The method proposed in [14] uses an external surface to rotate the object. This approach does not control the gripping force, while the majority of the works on pivoting exploit the control of the gripper's fingers for a successful outcome of the pivoting action.

An example in which the authors exploit the gripping force to apply a dissipative torque at the pivoting point, hence controlling the rotational motion of the object, is the work described in [15] and extended in [16]. This work focuses on *swing-up* motions, so the plane of rotation of the object coincides with the vertical direction. The proposed solution uses an energy-based controller to move the object from one position to another with higher potential energy. Apart from controlling the gripping force, this approach moves the arm to provide inertial forces sufficient to counteract the gravity.

While swing-up motions imply that the object moves against the gravity acceleration, the work described in [17] and [18] focuses on motions that exploit the gravity to reorient an object from a position of higher potential energy to one with lower potential energy. The gripper does not move, but it adjusts the distance between the fingers to

control the torsional friction and successfully reorient the object at the desired angle.

The previous approaches using controlled friction rely on fast feedback and high-frequency controllability of the gripper's fingers. However, many commercial robots are equipped with parallel grippers that have a control frequency of at most 10 Hz, and most of the commercial cameras provide images at 30 fps, which also increases the challenge of accurately tracking the object when it is rotating at a high speed. Therefore, we use a method that does not rely on high-performing hardware to successfully achieve in-hand manipulation.

The method proposed in [19] addresses the possible lack of high-performing hardware by modeling the delays in actuation and the noise in the sensors and it exploits Deep Reinforcement Learning to generate optimal actions. However, the pivoting action requires the robot to move multiple times back and forth for the object to be correctly reoriented, and this method requires tracking the object while it moves at a fast speed.

The previous approaches on pivoting considered it a separate action to be performed by the robot manipulator. Conversely, our work focuses on integrating the pivoting action with the robot's motion to obtain the successful reorientation of the object together with the execution of an overall task.

Moreover, the pivoting solution that we use does not pose constraints on the orientation of the object's plane of rotation nor on the direction of the rotation itself. Hence, it can be integrated with a generic motion of the robot and it is suitable for directly executing a task without seeing the reorientation as a separate step.

III. PROBLEM DEFINITION

This section provides first a description of the overall problem of pivoting an object while the robot is moving, and then it provides a formulation of the pivoting problem, which introduces the notation used in the following sections.

A. Integration Problem

The robot's task consists of picking up an object and moving it to a different place, to be used as a tool or to be placed back in a different configuration. In both cases, the object has to be held by the gripper in a particular pose. However, this pose is different from the one resulting after the initial grasp.

The overall problem consists of finding a feasible trajectory to move the gripper from its initial pose P_0 to the desired final pose P_f , and at the same time move the object inside the gripper from the initial orientation θ_0 to the desired orientation θ_d .

To solve this problem, we perform a geometric path planning off-line, and the constraints coming from the pivoting action are imposed at a later stage, modifying the timing law of the trajectory while maintaining the collision-free geometric path.

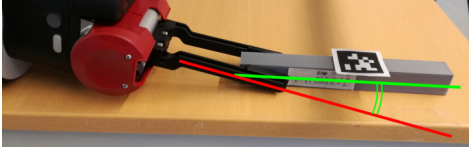


Fig. 2: An example of grasping in which it is impossible to achieve the desired orientation of the object of 0° with respect to the gripper, shown in red, due to the contact between the robot and the table.

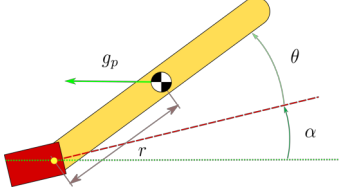


Fig. 3: Planar representation of an object rotating within a parallel gripper.

While moving the end-effector from the initial configuration to the desired one, the robot moves along a trajectory that takes into account obstacles in the environment and possible additional constraints. The pivoting solution that we choose forces the rotation of the object to happen only when the end-effector stops. Hence, during the motion of the robot manipulator the pose of the object within the gripper remains constant, simplifying the problem of finding a feasible trajectory in the presence of obstacles. In fact, the collisions between the object and the obstacles can be checked more easily, without a prediction of possible configurations or an unnecessarily large bounding box that would be required if the object were moving.

B. Pivoting Problem Formulation

We consider a system composed of a robot manipulator with a parallel gripper at the end-effector. The robot grasps an object with an initial angle θ_0 with respect to the gripper, which differs from the desired angle θ_d necessary to execute the final task. This difference can be due to joint limits or environmental constraints, as shown in the example in Fig. 2. We assume that the fingers grasp the object sufficiently distanced from the center of mass, so that the object can rotate. The shape of the object can be irregular, as long as it allows the rotational motion around the pivoting point without collisions with the gripper.

The goal is to reach the desired final pose for the robot manipulator so that the orientation of the object inside the gripper changes into the desired one after the robot's motion.

We assume that the dimension of the object and its mass and inertia can be measured or inferred from the available sensors. In case the friction coefficients between the object and the gripper's fingers are unknown, it is possible to follow the approach proposed in [11] to estimate them before performing the final task. This approach updates the model according to the mismatch between expected and measured final angle. It can be easily integrated with our method of split path described later in section V-B.3.

Fig. 3 represents the object, grasped by a parallel gripper, that has to rotate around a pivoting point. Assuming that the

robot is not moving, the dynamic model of the rotation of the object is:

$$(I + mr^2)\ddot{\theta} - mg_p r \sin(\theta + \alpha) = \tau_f, \quad (1)$$

in which I is the inertia of the object, m its mass and r the distance between its center of mass and the pivoting point; θ is the angle of the object with respect to the gripper and $\ddot{\theta}$ is its angular acceleration; g_p is the component of the gravity acceleration in the plane of rotation, which depends on the current pose of the gripper; α is the angle between the direction of the gravity and the gripper; τ_f is the torsional friction at the pivoting point.

We assume that the contact area between the fingertips and the object is sufficiently small in relation to the distance to the object's center of mass so that the contact can be described as a single point and this pivoting point is always well defined. With this assumption, given the model in (1), irregular shapes of objects or different pivoting points do not affect the pivoting action as long as the parameters I , m , r and the friction coefficients are known.

We use the Coulomb friction model to describe the static friction τ_s when the object is not moving:

$$|t_s| \leq \gamma f_n, \quad (2)$$

where γ is the coefficient of friction and f_n is the normal force applied by the gripper's fingers to the object.

When the object is moving, we choose to model the torsional friction as Coulomb and viscous friction [20]:

$$\tau_f = -\mu\dot{\theta} - \sigma \text{sgn}(\dot{\theta})f_n, \quad (3)$$

where μ and σ are friction coefficients, $\dot{\theta}$ is the angular velocity of the object and sgn is the signum function.

When the object starts rotating, switching from the model in (2) to the one in (3), we follow the approach proposed in [21] of defining a neighbor of $\dot{\theta}$ in which the normal force f_n balances the net torque, to avoid numerical singularities.

Since many robots are not equipped with tactile sensors at the fingertips, (3) can be reformulated by expressing f_n as a function of the distance d between the two fingers. A possible solution, adopted in [11] and [17], assumes a linear deformation model:

$$f_n = k(d_0 - d), \quad (4)$$

in which k is a stiffness parameter and d_0 is the distance of zero deformation of the fingertips.

Assuming that the gripper is not moving, when the object starts its motion the evolution of the angle θ depends only on the initial angular velocity of the object $\dot{\theta}_0$ and on the distance between the gripper's fingers d . This distance is kept small enough to prevent translational slippage and allow only rotational slippage, as in [17].

The method we use for pivoting stops the motion of the gripper and opens its fingers only once to initiate the rotation of the object. Therefore, the problem consists in finding the values of $\dot{\theta}_0$ and d that allow the system to obtain the desired repositioning and are compatible with the desired motion execution of the robot manipulator for the overall task.

IV. PIVOTING METHOD

The pivoting method that we follow is the three-stages open loop pivoting described in [11], and we provide a brief summary of it below to clarify the integration with the robot manipulator's motion.

A. Three-stages Open Loop Pivoting

This method is divided into three stages, shown in Fig. 4:

1) *End-effector's velocity stage*: in this stage, the gripper moves at a certain velocity while holding the object firmly. This velocity causes the motion of the object in the third stage and determines the initial velocity $\dot{\theta}_0$ at which this motion starts.

2) *Finger distance stage*: in this stage, the gripper stops and opens the fingers. The distance at which it opens influences the torsional friction at the pivoting point, which in turn influences the motion of the object.

3) *Object's motion stage*: in this stage the object rotates around the pivoting point and it stops at a different orientation. This motion depends only on the actions taken at the previous stages.

B. Pivoting Control Actions

For the successful outcome of the pivoting action, at the end of the third stage the object should stop at the desired angle. This motion is influenced by the initial velocity of the object, by the distance between the gripper's fingers and by the gravity, which depends on the gripper's pose. Since during the third stage the gripper does not move, the gravity acceleration remains constant. Therefore, the pivoting problem can be defined as follows:

find the initial angular velocity $\dot{\theta}_0$ and the distance to open between the fingers d so that, according to the dynamic model in (1), $\theta_f = \theta_d$, with θ_f being the angle at which the object stops moving and θ_d being the desired angle.

To compute the optimal action pair $(d^*, \dot{\theta}_0^*)$, several solutions can be adopted, such as Reinforcement Learning algorithms or Dynamic Programming. Among the possible options, we use Q-Learning with the reward function R as:

$$R(\theta) = \begin{cases} 1 & \text{if } |\theta - \theta_d| \leq \delta \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

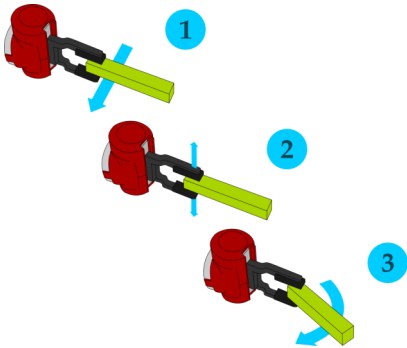


Fig. 4: The three separate stages of the open loop pivoting. First, the gripper and the object move together with the same velocity. Second, the gripper stops and opens the fingers. Third, the object rotates around the pivoting point and it reaches the desired angle.

in which δ is a tolerance margin for the desired angle θ_d .

As [11] highlights, this learning process is sufficiently fast to be executed on-line, i.e. while the robot is manipulating the object.

C. Coefficients Estimation

While it is simple to measure the mass and length of the object, the friction coefficients are more challenging to estimate. However, it is not required to have highly accurate values as long as the behavior predicted by (1) resembles the outcome of the real system. As described in [11], an estimate can be obtained from consecutive trials and minimizing the error between the predicted and the measured outcome.

V. INTEGRATION WITH THE ROBOT'S MOTION

In this section we propose a solution for the integration of the pivoting method with the motion of the robot manipulator in order to achieve a joint execution of both the repositioning of the object and the reaching of the desired end-effector's pose.

A. Problem Analysis and Proposed Method

A detailed description of the proposed method is discussed below, and it is summarized in Algorithms 1 and 2.

The initial angular velocity at which the object starts rotating around the pivoting point derives from the velocity of the end-effector in the instant before it stops. In fact, the center of mass of the object keeps moving with the same velocity while the gripper stops, but instead of proceeding on the same direction, its motion turns into a rotational motion due to the constraint posed by the gripper's fingers.

Assuming that the end-effector's final velocity is \mathbf{v} and that its projection on the plane of rotation of the object is \mathbf{v}_p , the initial angular velocity of the object around the pivoting point is:

$$\dot{\theta}_0 = \frac{\mathbf{v}_p \cdot \hat{\mathbf{r}}_{\perp}}{r}, \quad (6)$$

where $\hat{\mathbf{r}}_{\perp}$ is the unitary vector orthogonal to the direction that goes from the pivoting point to the center of mass of the object, pointing towards the positive direction of rotation, and \cdot is the scalar product.

The maximum velocity transmission is when \mathbf{v}_p and $\hat{\mathbf{r}}_{\perp}$ are parallel, but it is not always feasible to impose a certain direction of motion to the robot's end-effector due to possible obstacles in the environment and joint limits. Conversely, the minimum velocity transmission is when these two vectors are orthogonal, resulting in a null initial angular velocity and no rotation of the object.

Moreover, with ϕ being the angle between \mathbf{v}_p and $\hat{\mathbf{r}}_{\perp}$, the concordance between the signs of $\dot{\theta}_0$ and $\dot{\theta}_0^*$ is obtained only when $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$ if $\dot{\theta}_0^*$ is positive (i.e. $\theta_0 < \theta_d$), and when $-\pi < \phi < -\frac{\pi}{2}$ or $\frac{\pi}{2} < \phi < \pi$ if it is negative. Therefore, if the direction of motion disagrees with the desired angular velocity of the object, it is not possible to successfully achieve the desired pivoting action.

Given that the desired final pose P_f is known, the planar component of the gravity g_p and the angle α are computed

accordingly. The initial angle of the object θ_0 depends on the initial grasp. The optimal pivoting action (a^*, θ_0^*) is obtained with the method described in section IV given the previous quantities and the object's properties.

The object's properties can be included in the scene description S , which also includes the obstacles in the environment, and is used as input for our method.

From an initial pose P_0 of the robot's end-effector, at which the object is initially grasped, it is easy to compute a collision-free geometric path to the desired final pose P_f using standard planning algorithms. From this path, the direction of the velocity at a given point in the path can be derived by looking at the motion direction.

The final velocity direction $\hat{\mathbf{v}}$ is the one at which the center of mass of the grasped object continues to move after the gripper stops. However, the motion of the object is planar due to the constraint on the object imposed by the parallel gripper. Therefore, we consider only the components of the velocity that lie on the plane of rotation, i. e. a 2D vector $\hat{\mathbf{v}}_p$. The constraint in (6) can be rewritten as:

$$\dot{\theta}_0 = \frac{v_p \cos \phi}{r}, \quad (7)$$

in which v_p is the magnitude of the velocity component on the plane and ϕ is the angle between $\hat{\mathbf{v}}_p$ and $\hat{\mathbf{r}}_\perp$. Consequently, the desired magnitude of the velocity for guaranteeing the successful outcome of the pivoting action is:

$$v_p^* = \frac{r\dot{\theta}_0^*}{\cos \phi}. \quad (8)$$

Let $\hat{\mathbf{v}}_e = \mathbf{R}\hat{\mathbf{v}}$ be the 3D unitary vector describing the direction of the final velocity in the end-effector's reference frame, whose orientation is expressed by the rotation matrix \mathbf{R} . This frame is taken so that the xy plane coincides with the plane of rotation of the object. With $\hat{\mathbf{v}}_e$ denoting the 2D vector with the x and y components of $\hat{\mathbf{v}}_e$, then:

$$\hat{\mathbf{v}}_p = \frac{\hat{\mathbf{v}}_e}{\|\hat{\mathbf{v}}_e\|}. \quad (9)$$

By forcing the magnitude of the planar component of the velocity to be v_p^* from (8), the desired 3D velocity vector at the end of the computed path is:

$$\mathbf{v}^* = \frac{v_p^*}{\|\hat{\mathbf{v}}_e\|} \hat{\mathbf{v}}. \quad (10)$$

Therefore, given the geometric path, the trajectory, which comprehends the timing law, is obtained taking into account the velocity constraint in (10).

B. Additional Constraints

While generating a trajectory along a given geometric path, our method can face a situation in which the dynamics generated by the robot's motion are highly suboptimal or insufficient for the successful outcome of the pivoting task, or in which the robot is not able to execute the generated trajectory. We propose the following additional solutions to tackle the possible problems:

1) *Constrain the acceptable directions:* We have already mentioned that the pivoting action becomes unfeasible in case of a discordance between the direction of the gripper's velocity and the desired direction of rotation. Moreover, the transfer of the velocity from the gripper to the rotation of the object decreases as ϕ goes closer to $\pm \frac{\pi}{2}$, which means that to obtain the same $\dot{\theta}_0^*$ the robot has to move faster. Therefore, when planning, it is preferable to include a maximum and a minimum tolerable angle to improve the performances and minimize the risk of obtaining unfeasible velocities for the robot manipulator.

2) *Introduce likelihood of acceptable dynamics:* An initial generic motion direction can be derived by simply looking at the vector from P_0 to P_f . If this direction fully disagrees with the desired direction of rotation of the object, i.e. the estimated ϕ derived from this direction leads to $\cos \phi < 0$ when $\dot{\theta}_0^* \geq 0$ or to $\cos \phi \geq 0$ when $\dot{\theta}_0^* < 0$, the pivoting action becomes unfeasible. In many cases, it is possible to solve this problem by executing a rotation of the gripper around the final joint, but this rotation affects the final pose of the gripper that will be different from the desired one, which is especially important if the gripper has additional sensors (e.g. a camera) that will end in a wrong configuration. Therefore, we suggest to add a waypoint in the planned path so that the direction of motion from it to the final point satisfies the concordance constraint. More specifically, this direction is the direction $\hat{\mathbf{u}}$ from P_f to the waypoint P_w , with component $\hat{\mathbf{u}}_p$ in the plane of rotation in P_f , so that:

$$\begin{aligned} \hat{\mathbf{u}}_p &= \underset{\hat{\mathbf{u}}_p}{\operatorname{argmax}} |(-\hat{\mathbf{u}}_p) \cdot \hat{\mathbf{r}}_\perp| \\ \text{s.t.} \quad & -\frac{\pi}{2} < \phi' < \frac{\pi}{2} \quad \text{if } \theta_0 < \theta_d \\ & -\pi < \phi' < -\frac{\pi}{2} \text{ or } \frac{\pi}{2} < \phi' < \pi \quad \text{otherwise,} \end{aligned} \quad (11)$$

in which ϕ' is the angle between $-\hat{\mathbf{u}}_p$ and $\hat{\mathbf{r}}_\perp$. Therefore, the waypoint is chosen by translating the final pose of a distance h along the direction $\hat{\mathbf{u}}$. According to the chosen motion planner and to the setup, this waypoint can be seen as a soft constraint, without forcing the robot manipulator to plan exactly through that end-effector's position and orientation.

3) *Split the path for Collision-free trajectories:* Since it is not always possible to insert waypoints that enable the successful planning of a collision-free trajectory, we include an additional step for successfully obtaining the pivoting action when the added constraints are not sufficient or infeasible.

Once a collision free path is obtained, and adding constraints does not introduce a final velocity direction that guarantees a good transfer of the velocity from the gripper to the rotation of the object, we propose to search along the whole path for a velocity direction that instead satisfies the concordance constraint. That is, find the path point P_i so that the velocity direction $\hat{\mathbf{v}}_i$ at this point, with $\hat{\mathbf{v}}_{i,p}$ its planar projection on the plane of rotation in P_i , is so that:

$$\hat{\mathbf{v}}_{i,p} = \underset{\hat{\mathbf{v}}_{i,p}}{\operatorname{argmax}} |\hat{\mathbf{v}}_{i,p} \cdot \hat{\mathbf{r}}_\perp|, \quad (12)$$

Algorithm 1: execute_task

Input: S, P_0, P_f, θ_d

```

1 grasp the object
2 observe  $\theta_0$ 
3  $\hat{d} \leftarrow$  direction from  $P_0$  to  $P_f$ 
4 if  $(\hat{d} \cdot \hat{r}_\perp > 0 \wedge \theta_0 \leq \theta_d) \vee (\hat{d} \cdot \hat{r}_\perp \leq 0 \wedge \theta_0 > \theta_d)$  then
5   |  $w \leftarrow$  none
6   | trajectory  $\leftarrow$  compute_trajectory( $S, P_0, P_f, \theta_0, \theta_d, w$ )
7 end
8 else
9   |  $w \leftarrow P_w$  satisfying (11)
10  | trajectory  $\leftarrow$  compute_trajectory( $S, P_0, P_f, \theta_0, \theta_d, w$ )
11 end
12 if trajectory is none then
13   |  $w' \leftarrow w \cup P'_w \neq P_w$  satisfying (11)
14   | trajectory  $\leftarrow$  compute_trajectory( $S, P_0, P_f, \theta_0, \theta_d, w'$ )
15 end
16 execute trajectory
17 observe  $\theta_f$ 
18 return
```

and satisfies the same constraints of (11) given ϕ' as the angle between $\hat{v}_{i,p}$ and \hat{r}_\perp . However, this solution introduces the need for replanning the final segment because, after the pivoting action, the configuration of the object within the gripper is different. Moreover, it is important to plan the pivoting action considering the pose of the gripper at this point P_i , i.e. compute the correct values of g_p and α , which differ from the ones at the final goal pose.

As mentioned in section III-B, this method can be exploited also in case of uncertainty in the model's coefficients: the path is split multiple times to measure the final outcome of the pivoting action and the friction coefficients are updated until the model is accurate enough, while moving towards the goal pose. As described in [11], about 8 steps would be required to obtain a good estimate. Then, the computed action will lead to the desired angle.

C. Hardware Limitations

Given that the planned trajectory has to be executed on a real system, it is important that it satisfies the limitations imposed by the chosen hardware, such as joint velocities and acceleration limits. Therefore, the obtained trajectory must pass a feasibility check, otherwise it is discarded and a new path is preferred, as shown in Algorithm 2. However, the occurrence of this situation is extremely infrequent. In fact, it happens mostly if the final velocity \mathbf{v}^* exceeds the limits. This velocity is limited by the maximum allowed initial angular velocity of the object output by the three-stages pivoting. By safely choosing this maximum velocity, and by adding a requirement for the angle ϕ above the minimum, the event of finding a solution that exceeds the actuation limits is highly unlikely.

Another possible problem posed by the real system regards the un-modeled effects that influence the initial rotation of the object around the pivoting point. We assumed no dispersion in the transmission of the velocity from the robot's end-effector to the object after the first stops and the latter begins the rotation. However, as explained in [11],

Algorithm 2: compute_trajectory

Input: $S, P_0, P_f, \theta_0, \theta_d$, waypoints

```

1 path  $\leftarrow$  path from  $P_0$  to  $P_f$ , with waypoints
2 compute  $g_p, \alpha$  at  $P_f$ 
3  $d^*, \theta_0^* \leftarrow$  three-stages pivoting action from  $\theta_0, \theta_d, g_p, \alpha$ 
4  $\hat{v} \leftarrow$  final velocity direction from path
5  $\hat{v}_e \leftarrow$  planar projection from  $\hat{v}, P_f$ 
6  $\mathbf{v}^* \leftarrow$  from (10)
7 trajectory  $\leftarrow$  impose timing law on path from  $\mathbf{v}^*$ 
8 if trajectory is feasible then
9   | return trajectory
10 end
11 else
12   |  $P_i \leftarrow$  from (12)
13   | if  $P_i \neq P_f \wedge |\hat{v}_{i,p} \cdot \hat{r}_\perp| > |\hat{v}_p \cdot \hat{r}_\perp|$  then
14     | trajectory  $\leftarrow$  compute_trajectory( $S, P_0, P_i, \theta_0, \theta_d$ ,
15       | none)
16     | path'  $\leftarrow$  path from  $P_i$  to  $P_f$ , with new object angle
17     | trajectory'  $\leftarrow$  trajectory  $\cup$  timing law on path'
18     | return trajectory'
19   | end
20   | return none
21 end
22 end
```

an estimate of this dispersion due to un-modeled effects is included in the estimate of the friction coefficients and is performed beforehand.

Additional hardware limitations include the gripper's controllability frequency and the camera's frame rate if vision is used to determine the object's pose. However, these limitation are already fully compensated by the choice of the pivoting strategy and by the integration method used for the execution of the overall task.

VI. EXPERIMENTS

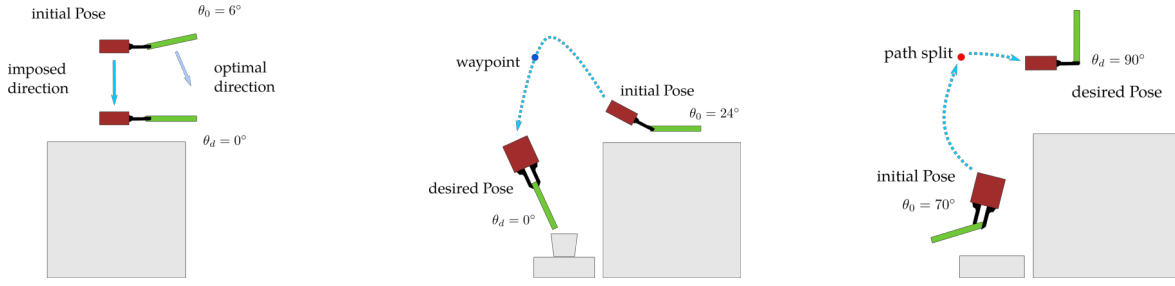
We tested the proposed approach for integrating pivoting and robot motion using a Baxter robot.

A. General setup

The parameters of the manipulated object are as follows: I is 0.000057248 kg m², m is 0.024 kg, r is 0.084 m, d_0 is 0.0189 m, μ is 0.00568 kg m² s⁻¹, σk is 11.976 N. Since we were using the friction model derived from (3) and (4), we estimated a unique friction coefficient, σk , that includes the Coulomb friction coefficient and the stiffness parameter.

We used slightly deformable fingertips for changing the rotational friction by adjusting the distance between the fingers; fingertips appositely designed for pivoting, such as [9], would also integrate well with our method.

For the Q-Learning implementation used by the three-stages pivoting we used pybrain [22]. We measured the gripper's accuracy to be roughly 0.0005 m when commanded to a target distance between the fingers. We selected possible distances for the pivoting action to be between 0.0171 m and 0.0187 m with a discretization step of 0.0004. The value of $\dot{\theta}_0$ ranged between 0 and 21 rad/s; the velocity direction in the learning process is chosen to be always positive, and the reference of the angles is modified accordingly. The learning



(a) **1st experiment:** the robot holds the object at an angle $\theta_0 = 6^\circ$ and it is forced to move along a direction that does not ensure the best velocity transfer to the object, in order to achieve the desired angle $\theta_d = 0^\circ$.

(b) **2nd experiment:** the robot picks up the object from the table at an angle $\theta_0 = 24^\circ$ and it has to move to the desired pose, at which the desired orientation is $\theta_d = 0^\circ$. A waypoint is added to improve the repositioning.

(c) **3rd experiment:** the robot picks up the object at an angle $\theta_0 = 70^\circ$ and it has to reach the desired pose, at which $\theta_d = 90^\circ$. A waypoint cannot be added, and the repositioning is performed in the middle of the planned path.

Fig. 5: A representation of the experiments. The robot's gripper is represented in red and the object in green. The objects in grey represent the obstacles.

tolerance was set to ± 0.05 radians ($\pm 2.86^\circ$). This tolerance allowed us to have a final angle close to the target one and at the same time compensate for small errors in actuation and in perception. We chose the acceptable range of $|\cos \phi|$ for a good velocity transmission to be within 0.5 and 1, with the sign depending on the desired direction of motion, hence constraining the range of ϕ as suggested in section V-B.1.

For detecting the target object and its configuration with respect to the gripper we used April tags and a Kinect 2. The accuracy of the detection of the angle θ was expected to be around 2° , due to the imprecision of the camera to robot calibration. For path planning we used the OMPL library, in particular the RRT connect planner.

B. Robot Experiments

We performed different experiments to test the method under different conditions for the pivoting action.

1) *First Experiment:* In this experiment, shown in Fig. 5a, to test the behavior of the pivoting action under non-optimal conditions, the path was imposed to be a straight line downwards in the vertical direction. This is a case in which the direction, despite being suboptimal, is still good for reorienting the object, so no deviation from the given path is necessary. The initial angle at which the object was grasped was $\theta_0 = 6^\circ$, and the target angle was set to $\theta_d = 0^\circ$. The motion of the end-effector was constrained at a constant orientation in which the gripper's fingers were parallel to the ground. The influence of gravity was given by $g_p = 9.8 \text{ m/s}^2$ and $\alpha = -90^\circ$. Since the robot was moving in the same direction of the desired rotation of the object, there was the risk of overshooting the desired angle. However, thanks to the combination of a slow motion coming from the timing law computed for the given path, and a tight gripper opening (1.79 cm) chosen as optimal actions, the object was successfully repositioned within the acceptable tolerance, ending with an orientation of $\theta_f = 1.9^\circ$.

Once we established the correctness of the integration strategy, we proceeded to evaluate the performances of our proposed method by applying it on complete robot tasks.

2) *Second Experiment:* In this experiment, the robot had to grasp an object and place it in a cup. Since the general

direction from the initial gripper's pose to the final one was found to have an angle ϕ that was out of the tolerable range, our system added a waypoint above the cup to improve the velocity transmission from the gripper to the object. We chose a distance $h = 0.3 \text{ m}$ to select the waypoint pose P_w , which was sent to the motion planner as soft constraint. Therefore, this experiment shows an example of the solution discussed in section V-B.2. A scheme of this particular setup is shown in Fig. 5b. This second experiment was repeated multiple times (12) to measure the percentage of successful outcomes of the combination of pivoting with motion planning.

We constrained the motion of placing the object not to be vertical, in order to distinguish a successful pivoting action from a reorientation of the object simply due to the static equilibrium pose due to the gravity. Therefore, the influence of the gravity in the goal pose was given by $g_p = 9.7 \text{ m/s}^2$ and $\alpha = -169^\circ$. With this final configuration, the static equilibrium point of the object would lead to $\theta_f = -11^\circ$, while the desired angle was $\theta_d = 0^\circ$.

Since the robot regrasped the object at every retrieval, the initial angle slightly varied and ranged between 23.2° and 25.6° . This regrasping affected also the position of the pivoting point, i.e. the value of r , but this small variation did not affect the successful outcome of the reorientation.

The pivoting experiment was successful in 83% of the attempts. We considered a success a reorientation of the object within $\pm 2.86^\circ$ from the desired angle, which is the same tolerance used in the learning. However, failed experiments were still able to successfully execute the overall task of inserting the object in the cup, because the reorientation error was still contained within a small boundary of 5° , which was sufficient for the insertion of the object in this particular task. These errors are due to non negligible errors and delays in actuation which lead to a mismatch between the executed trajectory and the commanded one, as well as errors in the opening of the fingers due to the gripper's low accuracy of 0.5 mm. The results of this experiment are shown in Fig. 6, in terms of reorientation error.

3) *Third Experiment:* We designed the third experiment to test the behavior of the system in case of a solution that

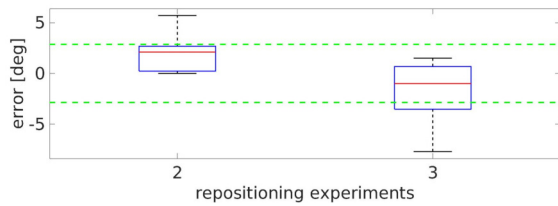


Fig. 6: Box plot of the error of the second and third experiment. The green lines mark the values of $\pm 2.86^\circ$. The red line represents the median and the whiskers represent the minimum and the maximum.

requires a split of the path instead of adding a waypoint to obtain a proper velocity for pivoting, as described in section V-B.3. In this experiment, the desired orientation of the object was 90° , and after the repositioning the object was supposed to be placed vertically on the table. The desired orientation was not achievable when the robot grasped the object due to a configuration close to joint limits. The initial orientation of the object varied between 70.1° and 72.6° in the 14 trials.

In this setup, the general direction of motion was suitable for the repositioning, however the final velocity direction of the planned path was not. A waypoint could not be added due to a configuration of the robot that would lead to self-collisions, therefore the pivoting action was performed in the middle of the planned path, in which the velocity direction was more suitable. This experiment is summarized in Fig. 5c.

Due to the randomized path planner, the intermediate pose at which the pivoting action was executed was not always the same. However, a successful pivoting action was always executed in a vertical plane, i.e. $g_p \approx 9.8$, and the value of α varied around -70° . We found that this experiment was more challenging than the previous one, and the repositioning was successful 71% of the times. The result is shown in Fig. 6. The challenge was due to the pivoting action requiring a fast motion to contrast the gravity acceleration, starting from a pose in which the robot was close to joint limits. Despite discarding trajectories that did not respect the joint limits, the actuation of the robot manipulator was more subject to errors, and small mismatches between the fingers opening were not negligible for the successful outcome of the pivoting action. However, unsuccessful reorientations can be overcome by running our algorithm a second time with a new starting pose and initial angle.

VII. CONCLUSIONS

In this work we proposed a method for integrating in-hand repositioning of objects using a pivoting strategy with the desired motion of the robot manipulator, to reduce the time spent in the execution of a complex task and enhance the system's efficiency. We have shown how to achieve a successful pivoting action together with the robot's motion using standard hardware platforms.

As future work, we plan to improve the constrained path planning to modify the orientation of the gripper in the pose chosen for pivoting, so that the influence of gravity can be controlled when it is strongly against the desired motion.

REFERENCES

- [1] P. Tournassoud, T. Lozano-Perez, and E. Mazer, "Regrasping," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, Mar 1987, pp. 1924–1928.
- [2] R. Paolini and M. T. Mason, "Data-driven statistical modeling of a cube regrasp," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 2554–2560.
- [3] A. Bicchi, "Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 652–662, Dec 2000.
- [4] K. Or, M. Tomura, A. Schmitz, S. Funabashi, and S. Sugano, "Interpolation control posture design for in-hand manipulation," in *2015 IEEE/SICE International Symposium on System Integration (SII)*, Dec 2015, pp. 187–192.
- [5] B. Sundaralingam and T. Hermans, "Relaxed-rigidity constraints: In-grasp manipulation using purely kinematic trajectory optimization," in *Robotics: Science and Systems*, 2017.
- [6] M. V. Liarokapis and A. M. Dollar, "Learning task-specific models for dexterous, in-hand manipulation with simple, adaptive robot hands," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 2534–2541.
- [7] N. Rahman, L. Carbonari, M. D'Imperio, C. Canali, D. G. Caldwell, and F. Cannella, "A dexterous gripper for in-hand manipulation," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2016, pp. 377–382.
- [8] W. G. Birchler, A. M. Dollar, and N. Rojas, "A two-fingered robot gripper with large object reorientation range," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3453–3460.
- [9] H. Terasaki and T. Hasegawa, "Motion planning of intelligent manipulation by a parallel two-fingered gripper equipped with a simple rotating mechanism," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 207–219, Apr 1998.
- [10] N. C. Daffe, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhrbringer, "Extrinsic dexterity: In-hand manipulation with external forces," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1578–1585.
- [11] S. Cruciani and C. Smith, "In-hand manipulation using three-stages open loop pivoting," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 1244–1251.
- [12] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, no. 5, pp. 1048–1066, Nov. 1993.
- [13] M. Cefalo and G. Oriolo, "Task-constrained motion planning for underactuated robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2965–2970.
- [14] A. Holladay, R. Paolini, and M. T. Mason, "A general framework for open-loop pivoting," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3675–3681.
- [15] A. Sintov and A. Shapiro, "Swing-up regrasping algorithm using energy control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4888–4893.
- [16] A. Sintov, O. Tsilil, and A. Shapiro, "Robotic swing-up regrasping manipulation based on the impulse-momentum approach and clqr control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1079–1090, Oct 2016.
- [17] F. E. Viña, Y. Karayiannidis, K. Pauwels, C. Smith, and D. Kragic, "In-hand manipulation using gravity and controlled slip," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 5636–5641.
- [18] F. E. Viña, Y. Karayiannidis, C. Smith, and D. Kragic, "Adaptive control for pivoting with visual and tactile feedback," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 399–406.
- [19] R. Antonova, S. Cruciani, C. Smith, and D. Kragic, "Reinforcement learning for pivoting task," *arXiv preprint arXiv:1703.00472*, 2017.
- [20] H. Olsson, K. J. Strm, M. Gfvert, C. C. D. Wit, and P. Lischinsky, "Friction models and friction compensation," *Eur. J. Control*, p. 176, 1998.
- [21] D. Karnopp, "Computer simulation of stick-slip friction in mechanical dynamic systems," *J. Dyn. Syst. Meas. Control.*, vol. 107, no. 1, pp. 100–103, 1985.
- [22] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," *Journal of Machine Learning Research*, vol. 11, pp. 743–746, 2010.